PLR

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/748,165	12/27/2000	Nigel C. Paver	219.39313X00	5048

20457 7590 10/26/2004

ANTONELLI, TERRY, STOUT & KRAUS, LLP
1300 NORTH SEVENTEENTH STREET
SUITE 1800
ARLINGTON, VA 22209-9889

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 10/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/748,165	**Applicant(s)** PAVER, NIGEL C.
	Examiner David J. Huisman	**Art Unit** 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 December 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-19 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Extension of Time and Amendment as received on 9/13/2004.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The examiner recommends amending the title to more accurately reflect the content of the independent claims, i.e., the combination of arithmetic flags.

4. On page 13, lines 2-10, steps 520 and 540 should be explained more clearly. It is not clear to the examiner in step 520 how 4 bits of the destination register (31:28) can be set to 12 bits (which come from nibbles 0-2, where each nibble comprises 4 bits). Also, in step 540, it is unclear how 8 bits of the destination register (31:24) can be set to 16 bits (which come from bytes 0-1, where each byte comprises 8 bits). The examiner is assuming that the applicant had meant to say that the destination register bits, for instance, bits 31-28, are set to either nibble 0, nibble 1, or nibble 2, but not nibbles 0-2. However, the specification should illustrate this concept more clearly. If the examiner is incorrect, then applicant should notify the examiner on the proper implementation.

Drawings

5. The drawings are objected to because of the following minor informalities: Regarding Fig.6, in step 520, it is unclear how 4 bits of the destination register (31:28) can be set to 12 bits (which come from nibbles 0-2, where each nibble comprises 4 bits). Also, in step 540, it is unclear how 8 bits of the destination register (31:24) can be set to 16 bits (which come from bytes 0-1, where each byte comprises 8 bits). The drawings should illustrate this concept more clearly. A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

6. Claims 1, 7, 12, and 17 are objected to because the examiner is unclear on why "the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items." Shouldn't the arithmetic flags represent the results obtained from mathematical operations performed on the data items instead of representing the data items themselves? For instance, if an ADD instruction is to add two data items, the arithmetic flags would represent the status of the addition result, and not the data items themselves. Appropriate correction is required.

Applicant argues against this objection but it is still unclear to the examiner. Normally, when arithmetic flags are used, the flag will represent the result of some operation, and not the operands. For example, take an ADD instruction which adds two operands. The examiner asserts that the arithmetic flag will normally be associated with the result. That is, if 1 and -1 are added, the result is zero, and a Z flag may be set to represent that the result is 0. On the other hand, if -1 and -1 are added, the result would be -2, and an N flag may be set to represent that the result is negative. Applicant is not claiming that the flag is associated with the result of the operation, but instead with the operands involved in the operation. More specifically, applicant claims "wherein the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items." From this, the flags represent the status of data which was operated upon, i.e., the status of the operands, and not the results. Again, applicant should clarify his/her position.

7. Claim 1 recites the limitation "the status" in line 6. There is insufficient antecedent basis for this limitation in the claim. Please replace "the status" with --a status--.

8. Claim 7 recites the limitation "the status" in line 4. There is insufficient antecedent basis for this limitation in the claim. Please replace "the status" with --a status--.

9. Claim 12 recites the limitation "the status" in line 4. There is insufficient antecedent basis for this limitation in the claim. Please replace "the status" with --a status--.

10. Claim 17 recites the limitation "the status" in line 4. There is insufficient antecedent basis for this limitation in the claim. Please replace "the status" with --a status--.

Maintained Rejections

11. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action for claims 1-19. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

Maintained Claim Rejections - 35 USC § 102

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

13. Claims 1-4, 6-9, and 17-19 are rejected under 35 U.S.C. 102(e) as being anticipated by Wilson, U.S. Patent No. 6,530,012.

14. Referring to claim 1, Wilson has taught a device for combining a plurality of arithmetic flags, comprising a combination function module (Fig.6) that:

a) examines a plurality of arithmetic flags (column 7, lines 48-51); and

b) determines field size of the plurality of arithmetic flags and based on the determination of the field size will combine the plurality of arithmetic flags into a single combined arithmetic flag variable, wherein the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items. It should be realized that a field size must first be determined. For purposes of this examination, "field size" in Wilson refers to the maximum number of unique condition codes that can be produced for a given instruction. A condition code is a 4-bit set of arithmetic flags (column 6, lines 37-42). From column 6, lines 43-45, when the SIMD data is one byte in size, it is determined that the field size is 8. More specifically, up to 8 unique condition codes will be produced. From column 6, lines 46-56, when the SIMD data is two bytes (half-word) in size, it is determined that the field size is 4. That is, 4 condition codes are produced and then duplicated. However, it is only possible for 4 unique codes to be produced. As a further example, from column 6, lines 58-65, when the SIMD data is 32 bits in size (word), the field size is determined to be 2. That is, 2 condition codes are produced and then duplicated multiple times. However, it is only possible for 2 unique codes to be produced. Finally, when the SIMD data size is 64 bits, the field size is 1, where 1 condition code is produced and duplicated multiple times. However, it is only possible for a single unique code to be produced. In the end, the plurality of arithmetic flags are stored in a single 64-bit arithmetic flag register (see Fig.5). These flags represent data item status (column 6, lines 1-5) after a mathematical operation is performed by the processor.

15. Referring to claim 2, Wilson has taught a device as described in claim 1. Wilson has further taught a condition check module that determines the status of the combined arithmetic flag variable and causes the processor to execute an appropriate operation based on the status. See column 8, lines 23-61.

16. Referring to claim 3, Wilson has taught a device as described in claim 1. Wilson has further taught that the field size is based on either a nibble, byte, half-word, or word. As described in the rejection of claim 1 above, and in column 6, lines 43-65, the field size is either a byte, half-word, word, or long-word. And, due to the applicant's use of the word "or" in the claim, Wilson is able to anticipate the claim since at least one of the claimed features has been taught.

17. Referring to claim 4, Wilson has taught a device as described in claim 3. Wilson has further taught that the plurality of arithmetic flags further comprise a negative data value, a zero

data value, a carry out occurrence in a data value, or an overflow condition in a data item in the plurality of data items. See column 6, lines 1-5.

18. Referring to claim 6, Wilson has taught a device as described in claim 2. Wilson has further taught that the status determined by the condition further comprises:

a) any data item has overflowed. If any one of the condition codes in the combined variable of Fig.7 has the V bit set (which represents overflow), then it is determined, by the CC Checker (Fig.7, component 50), that any data item has overflowed. See Table 1 in column 6 for further information.

b) any data item has not overflowed. If any one of the condition codes in the combined variable of Fig.7 has the V bit cleared (which represents no overflow), then it is determined, by the CC Checker (Fig.7, component 50), that any data item has not overflowed. See Table 1 in column 6 for further information.

c) any data item is positive or zero. If any one of the condition codes in the combined variable of Fig.7 has the N bit cleared (which represents a number not being negative), then it is determined, by the CC Checker (Fig.7, component 50), that any data item is positive or zero. See Table 1 in column 6 for further information.

d) any data item is negative. If any one of the condition codes in the combined variable of Fig.7 has the N bit set (which represents a negative number), then it is determined, by the CC Checker (Fig.7, component 50), that any data item is negative. See Table 1 in column 6 for further information.

e) any data item is zero. If any one of the condition codes in the combined variable of Fig.7 has the Z bit set (which represents zero), then it is determined, by the CC Checker (Fig.7, component 50), that any data item is zero. See Table 1 in column 6 for further information.

f) any data item is not zero. If any one of the condition codes in the combined variable of Fig.7 has the Z bit cleared (which represents a non-zero number), then it is determined, by the CC Checker (Fig.7, component 50), that any data item is not zero. See Table 1 in column 6 for further information.

g) any data item has a carry out. If any one of the condition codes in the combined variable of Fig.7 has the C bit set (which represents a carry), then it is determined, by the CC Checker (Fig.7, component 50), that any data item has a carry out. See Table 1 in column 6 for further information.

h) any data item does not have a carry out. If any one of the condition codes in the combined variable of Fig.7 has the C bit cleared (which represents no carry), then it is determined, by the CC Checker (Fig.7, component 50), that any data item does not have a carry out. See Table 1 in column 6 for further information.

i) all data items have overflowed. If all of the condition codes in the combined variable of Fig.7 has the V bit set (which represents overflow), then it is determined, by the CC Checker (Fig.7, component 50), that all data items have overflowed. See Table 1 in column 6 for further information.

j) all data items have not overflowed. If all of the condition codes in the combined variable of Fig.7 has the V bit cleared (which represents no overflow), then it is determined, by the CC

Checker (Fig.7, component 50), that all data items have not overflowed. See Table 1 in column 6 for further information.

k) all data items are positive or zero. If all of the condition codes in the combined variable of Fig.7 has the N bit cleared (which represents a number not being negative), then it is determined, by the CC Checker (Fig.7, component 50), that all data items are positive or zero. See Table 1 in column 6 for further information.

l) all data items are negative. If all of the condition codes in the combined variable of Fig.7 has the N bit set (which represents a negative number), then it is determined, by the CC Checker (Fig.7, component 50), that all data items are negative. See Table 1 in column 6 for further information.

m) all data items are zero. If all of the condition codes in the combined variable of Fig.7 has the Z bit set (which represents zero), then it is determined, by the CC Checker (Fig.7, component 50), that all data items are zero. See Table 1 in column 6 for further information.

n) all data items are not zero. If all of the condition codes in the combined variable of Fig.7 has the Z bit cleared (which represents a non-zero number), then it is determined, by the CC Checker (Fig.7, component 50), that all data items are not zero. See Table 1 in column 6 for further information.

o) all data items have a carry out. If all of the condition codes in the combined variable of Fig.7 has the C bit set (which represents a carry), then it is determined, by the CC Checker (Fig.7, component 50), that all data items have a carry out. See Table 1 in column 6 for further information.

p) all data items do not have a carry out. If all of the condition codes in the combined variable of Fig.7 has the C bit cleared (which represents no carry), then it is determined, by the CC Checker (Fig.7, component 50), that all data items do not have a carry out. See Table 1 in column 6 for further information.

19. Referring to claim 7, Wilson has taught a method of combining a plurality of arithmetic flags for presentation to a processor, comprising:

a) determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items. It should be realized that a field size must first be determined. For purposes of this examination, "field size" in Wilson refers to the maximum number of unique condition codes that can be produced for a given instruction. A condition code is a 4-bit set of arithmetic flags (column 6, lines 37-42). From column 6, lines 43-45, when the SIMD data is one byte in size, it is determined that the field size is 8. More specifically, up to 8 unique condition codes will be produced. From column 6, lines 46-56, when the SIMD data is two bytes (half-word) in size, it is determined that the field size is 4. That is, 4 condition codes are produced and then duplicated. However, it is only possible for 4 unique codes to be produced. As a further example, from column 6, lines 58-65, when the SIMD data is 32 bits in size (word), the field size is determined to be 2. That is, 2 condition codes are produced and then duplicated multiple times. However, it is only possible for 2 unique codes to be produced. Finally, when the SIMD data size is 64 bits, the field size is 1, where 1 condition code is produced and duplicated multiple times. However, it is only possible for a single unique code to be produced. These condition codes (arithmetic flags) represent data item status (column 6, lines 1-5) after a mathematical operation is performed by the processor.

b) extracting the plurality of arithmetic flags based on the field size. The "American Heritage® Dictionary of the English Language, Third Edition," 1992, has defined "extract" as "to derive or obtain from a source" and "to determine or calculate." As a result, Wilson's system will extract (derive or determine) the arithmetic flag values based on a mathematical operation performed by the processor. And, from the rejection of claim 7 above, the number of flags derived (extracted) depends on the field size.

c) combining the plurality of arithmetic flags based on a function selected when a combination process is selected. See Fig.6 and column 6, line 43, to column 7, line 6. Note that the functions would include the byte combination, the half-word combination, the word combination, and the long-word combination.

d) storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor. Note that the flags are stored in the destination register shown at the bottom of Fig.6.

20. Referring to claim 8, Wilson has taught a method as described in claim 7. Furthermore, claim 8 is rejected for the same reasons set forth in the rejection of claim 3.

21. Referring to claim 9, Wilson has taught a method as described in claim 8. Furthermore, claim 9 is rejected for the same reasons set forth in the rejection of claim 4.

22. Referring to claim 12, Wilson has taught an apparatus comprising a data storage medium for storing instructions (see Fig.1) when executed by a processor results in, comprising:

a) determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items. It should be realized that a field size must first be determined. For purposes of this examination, "field size" in Wilson refers to the maximum number of unique condition codes that can be produced for a given instruction. A condition code is a 4-bit set of arithmetic flags (column 6, lines 37-42). From column 6, lines 43-45, when the SIMD data is one byte in size, it is determined that the field size is 8. More specifically, up to 8 unique condition codes will be produced. From column 6, lines 46-56, when the SIMD data is two bytes (half-word) in size, it is determined that the field size is 4. That is, 4 condition codes are produced and then duplicated. However, it is only possible for 4 unique codes to be produced. As a further example, from column 6, lines 58-65, when the SIMD data is 32 bits in size (word), the field size is determined to be 2. That is, 2 condition codes are produced and then duplicated multiple times. However, it is only possible for 2 unique codes to be produced. Finally, when the SIMD data size is 64 bits, the field size is 1, where 1 condition code is produced and duplicated multiple times. However, it is only possible for a single unique code to be produced. These condition codes (arithmetic flags) represent data item status (column 6, lines 1-5) after a mathematical operation is performed by the processor.

b) extracting the plurality of arithmetic flags based on the field size. The "American Heritage® Dictionary of the English Language, Third Edition," 1992, has defined "extract" as "to derive or obtain from a source" and "to determine or calculate." As a result, Wilson's system will extract (derive or determine) the arithmetic flag values based on a mathematical operation performed by the processor. And, from the rejection of claim 7 above, the number of flags derived (extracted) depends on the field size.

c) combining the plurality of arithmetic flags based on a function selected when a combination process is selected. See Fig.6 and column 6, line 43, to column 7, line 6. Note that the functions would include the byte combination, the half-word combination, the word combination, and the long-word combination.

d) storing a result of the combining of the plurality of arithmetic flags in a destination register for access by the processor. Note that the flags are stored in the destination register shown at the bottom of Fig.6.

23. Referring to claim 13, Wilson has taught an apparatus as described in claim 12. Furthermore, claim 13 is rejected for the same reasons set forth in the rejection of claim 3.

24. Referring to claim 14, Wilson has taught an apparatus as described in claim 13. Furthermore, claim 14 is rejected for the same reasons set forth in the rejection of claim 4.

25. Referring to claim 17, Wilson has taught a method of extracting a plurality of arithmetic flags for presentation to a processor, comprising:

a) determining a field size of the plurality of arithmetic flags on which to base a combination process, wherein the plurality of arithmetic flags represent the status of a plurality of data items after a mathematical operation is performed by the processor on the plurality of data items. It should be realized that a field size must first be determined. For purposes of this examination, "field size" in Wilson refers to the maximum number of unique condition codes that can be produced for a given instruction. A condition code is a 4-bit set of arithmetic flags (column 6, lines 37-42). From column 6, lines 43-45, when the SIMD data is one byte in size, it is determined that the field size is 8. More specifically, up to 8 unique condition codes will be produced. From column 6, lines 46-56, when the SIMD data is two bytes (half-word) in size, it is determined that the field size is 4. That is, 4 condition codes are produced and then duplicated. However, it is only possible for 4 unique codes to be produced. As a further example, from column 6, lines 58-65, when the SIMD data is 32 bits in size (word), the field size is determined to be 2. That is, 2 condition codes are produced and then duplicated multiple times. However, it is only possible for 2 unique codes to be produced. Finally, when the SIMD data size is 64 bits, the field size is 1, where 1 condition code is produced and duplicated multiple times. However, it is only possible for a single unique code to be produced. These condition codes (arithmetic flags) represent data item status (column 6, lines 1-5) after a mathematical operation is performed by the processor.

b) extracting the plurality of arithmetic flags based on the field size. The "American Heritage® Dictionary of the English Language, Third Edition," 1992, has defined "extract" as "to derive or obtain from a source" and "to determine or calculate." As a result, Wilson's system will extract (derive or determine) the arithmetic flag values based on a mathematical operation performed by the processor. And, from the rejection of claim 7 above, the number of flags derived (extracted) depends on the field size.

c) storing a result of the extracting of the plurality of arithmetic flags in a destination register for access by the processor. Note that the flags are stored in the destination register shown at the bottom of Fig.6.

26. Referring to claim 18, Wilson has taught a method as described in claim 17. Furthermore, claim 18 is rejected for the same reasons set forth in the rejection of claim 3.

27. Referring to claim 19, Wilson has taught a method as described in claim 18. Furthermore, claim 19 is rejected for the same reasons set forth in the rejection of claim 4.

Maintained Claim Rejections - 35 USC § 103

28. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

29. Claims 5, 10-11, and 15-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilson, as applied above, in view of Bindloss et al., U.S. Patent No. 5,778,241 (herein referred to as Bindloss).

30. Referring to claim 5, Wilson has taught a device as described in claim 4. Wilson has not taught that the combination function module performs either an AND or an OR operation. However, Bindloss has taught the general concept of ANDing multiple condition codes together to produce an overall condition code. See Fig.3C and column 7, lines 32-36. Performing this operation would allow the system, for instance, to check if all data items have a carry. See column 4, lines 1-4. This should be realized because if 4 additions occur, each addition producing condition codes of 1101, 1011, 1100, and 1110, respectively (where the carry flag is the 0th bit), then by ANDing each code together, the final code is 1000. And, since the 0th bit is equal to 0, the system would recognize that a carry did not occur in all of the items. A person of ordinary skill in the art would have recognized that checking a single ANDed condition code would take less time than checking multiple condition codes. Therefore, to increase the

efficiency of the system, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Wilson such that the combination function module performs either an AND or an OR operation. Note that Bindloss' teaching of just an AND operation is sufficient enough to reject the claim since the applicant is using alternate language ("or").

31. Referring to claim 10, Wilson has taught a method as described in claim 9. Furthermore, claim 10 is rejected for the same reasons set forth in the rejection of claim 5.

32. Referring to claim 11, Wilson in view of Bindloss has taught a method as described in claim 10. Furthermore, claim 11 is rejected for the same reasons set forth in the rejection of claim 6.

33. Referring to claim 15, Wilson has taught an apparatus as described in claim 14. Furthermore, claim 15 is rejected for the same reasons set forth in the rejection of claim 5.

34. Referring to claim 16, Wilson has taught an apparatus as described in claim 15. Furthermore, claim 16 is rejected for the same reasons set forth in the rejection of claim 6.

Response to Arguments

35. Applicant's arguments filed on September 13, 2004, have been fully considered but they are not persuasive.

36. Applicant argues the novelty/rejection of claim 1 on page 25 of the remarks, in substance that:

> "Wilson does not describe a combination function module that "will combine the plurality of arithmetic flags into a single combined arithmetic flag variable."

37. These arguments are not found persuasive for the following reasons:

a) The examiner believes that a clear case was made as to how Wilson reads on applicant's claims. From Fig.6, it can be seen that an operation is performed on packed operands (in this case, addition), and condition codes for each packed result are generated which are combined and written to the large condition code register (bottom of Fig.6). This register is also shown in Fig.5. The examiner could find nothing in applicant's claims which distinguish applicant's combining function with Wilson's combining function. That it, applicant merely claims combining based on field size. Wilson clearly does this and is explained in column 6, line 37, to column 7, line 6. Based on the field size, certain numbers of codes are generated, copied, and then combined so that they may be written to a single destination register.

38. Applicant's arguments regarding claims 7, 12, and 17, which involve the assertion that Wilson does not teach combining a plurality of flags and/or a combining process, are responded to in the same fashion as above (for the argument regarding claim 1). More specifically, generating individual condition codes and putting them all together in a single register is a combination process in itself. Applicant will need to elaborate on his/her combination process to better distinguish itself from Wilson.

Conclusion

39. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
October 19, 2004

[illegible]
EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100